

# CONVERSIÓN MER A MODELO RELACIONAL

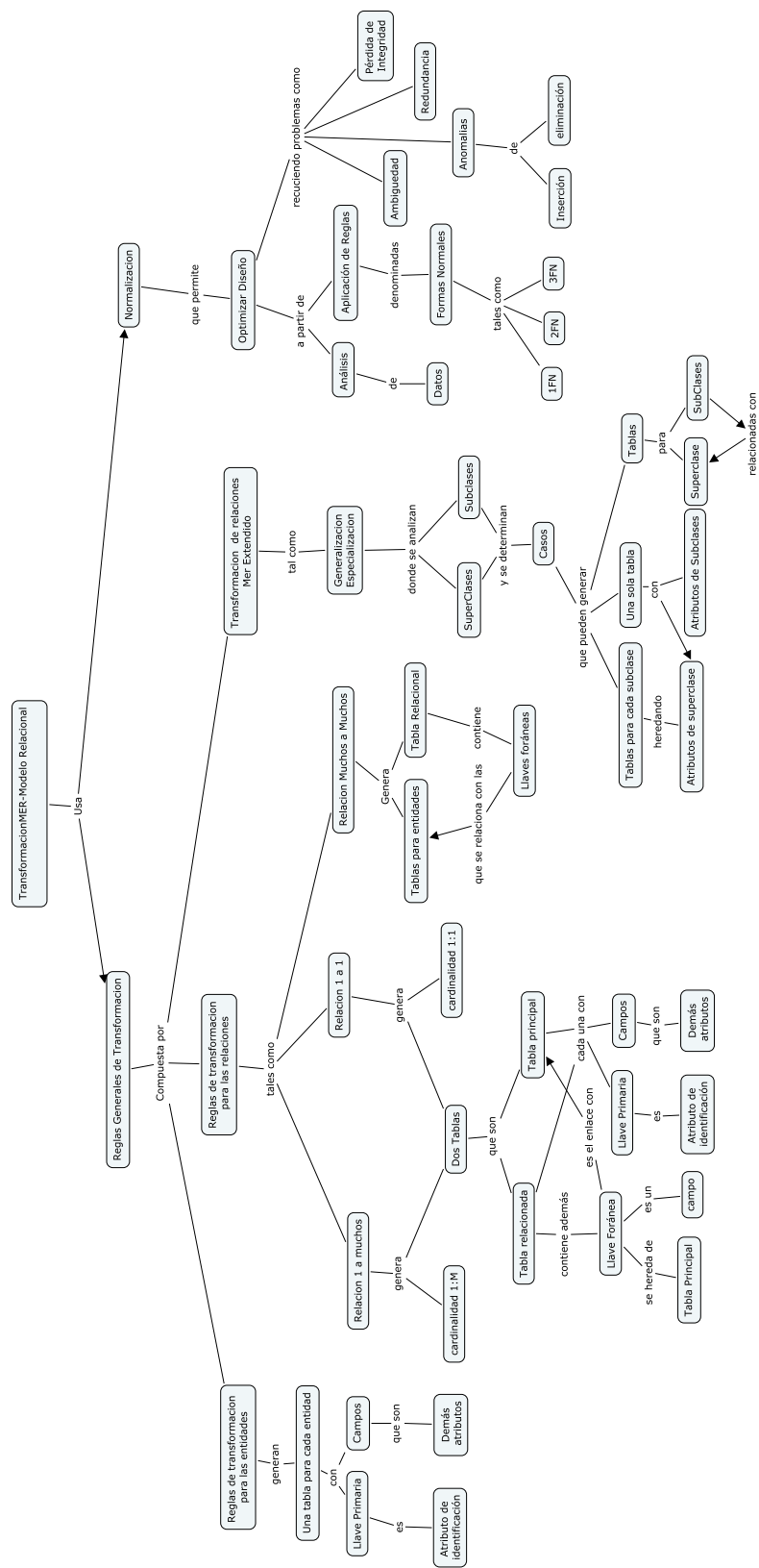
---

## Estructura de contenidos

1. INTRODUCCIÓN .....	3
2. TRANSFORMACIÓN MER A MODELO RELACIONAL.....	3
3. REGLAS GENERALES DE TRANSFORMACIÓN .....	5
3.1 Transformación para las entidades y atributos .....	5
3.2 Transformación para las Relaciones.....	7
4. TRANSFORMACIÓN MER- EXTENDIDO A RELACIONAL .....	11
5. CONSTRUCCIÓN TRANSFORMACIÓN MER- EXTENDIDO A RELACIONAL.....	14
5.1 Relación especialización y generalización .....	15
5.2 Relación Recursiva .....	20
6. NORMALIZACIÓN.....	23
6.1 Grados de Normalizacion.....	24



# Mapa conceptual CONVERSIÓN MER A MODELO RELACIONAL



### 1. INTRODUCCIÓN

Almacenar información es uno de los usos más frecuentes que una organización o empresa realiza día a día en un computador. La construcción e implementación de bases de datos permite que la organización pueda almacenar la información relevante del negocio a través de la construcción de una base de datos que se realiza por medio del modelado de datos. En este proceso están identificados tres modelos:

1. Modelo Entidad Relación (MER)
2. Modelo Relacional
3. Modelo Físico

En este material de consulta se presentará el proceso de construcción de un modelo relacional a partir de un MER, lo cual se denomina: **“Transformación del MER a Modelo Relacional”**.

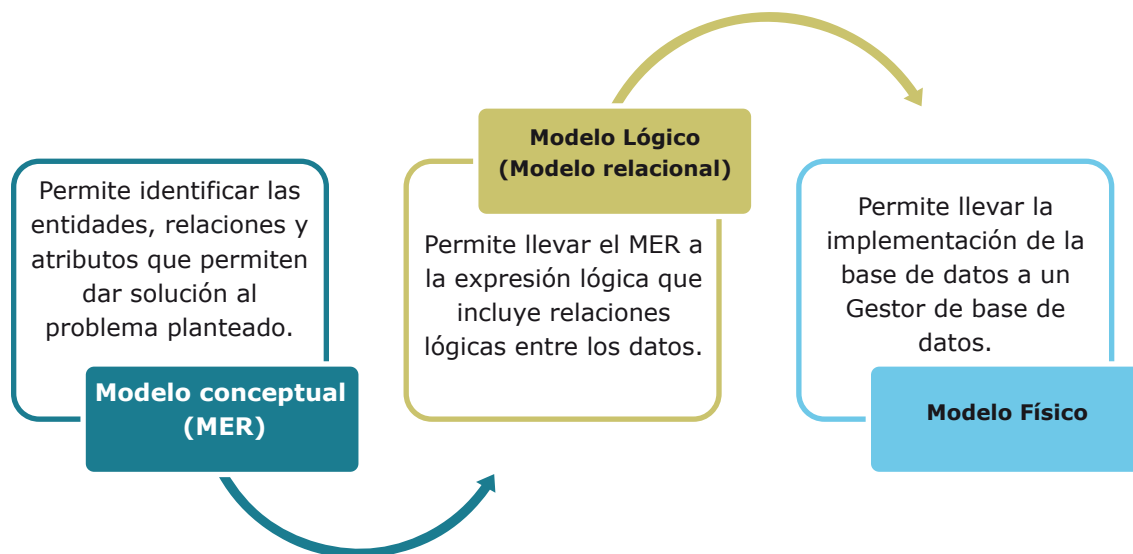
La intención es que el aprendiz logre desarrollar y despertar su interés en el modelado de datos, utilizando dicho modelo para la creación e implementación de una base de datos relacional.

### 2. TRANSFORMACIÓN MER A MODELO RELACIONAL

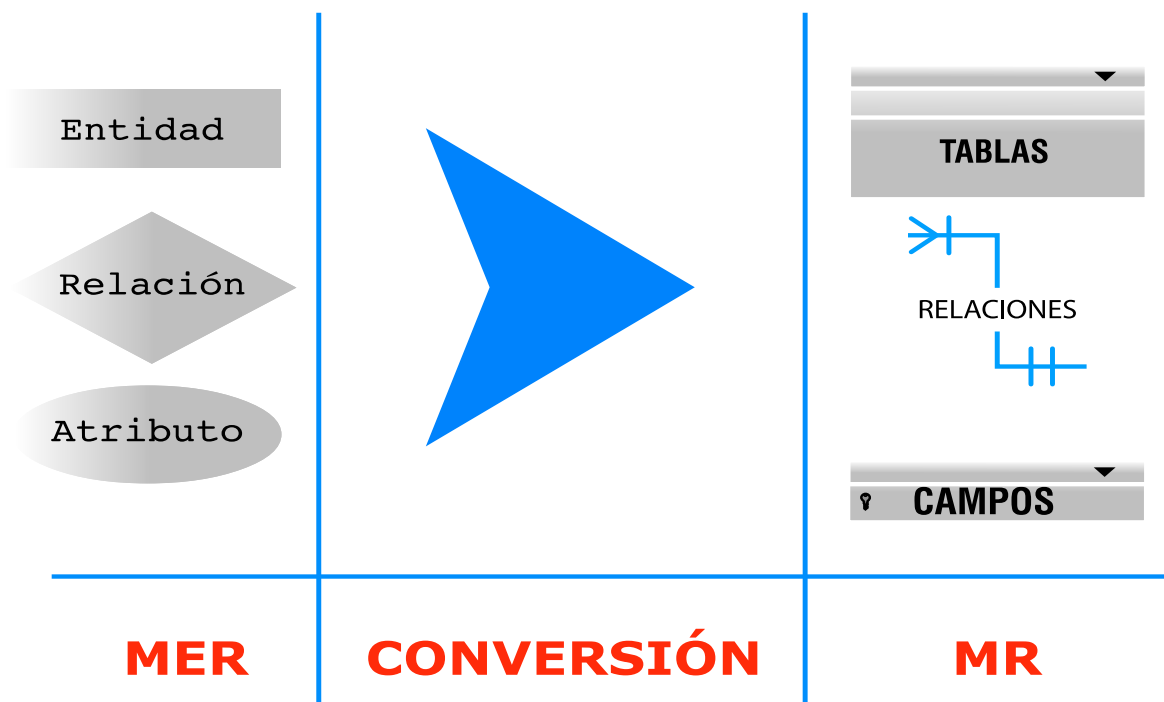
Uno de los pasos iniciales para la construcción de una base de datos es el modelado de ellos por medio de un Modelo Entidad Relación (MER), modelo que permite por medio de entidades, atributos y relaciones acercarse a una representación preliminar del diseño de una base de datos.

Luego de terminar un MER debe entrarse en el diseño de un modelo lógico, el cual es llamado modelo relacional. La forma como se logra desarrollar una base de datos relacional es a través del modelado y la metodología de diseño de una base de datos puede estar enfocada en los siguientes 3 pasos:



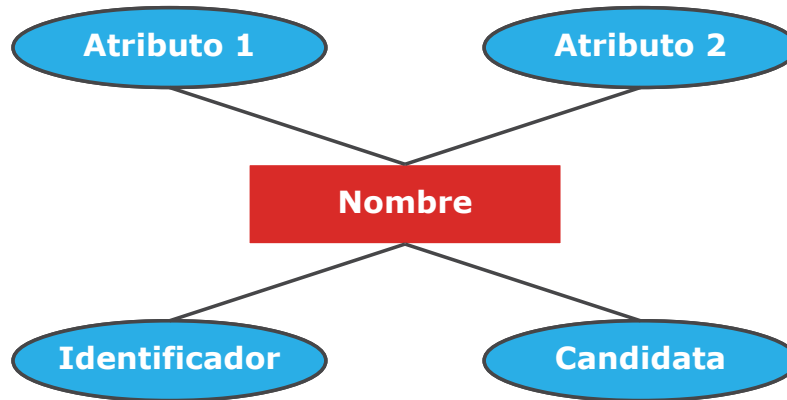


En estos momentos se abordará el proceso de la construcción de un modelo relacional a partir de un MER.



### 3. REGLAS GENERALES DE TRANSFORMACIÓN

La transformación de un Modelo Entidad Relación a Relacional, se puede expresar mediante un esquema relacional o un modelo relacional.



El esquema relacional consta de palabras y relaciones mediante flechas que indican la asociación de las entidades; el modelo relacional es un dibujo que representa las tablas, atributos, llaves primarias y llaves foráneas según la transformación realizada.

A continuación se explica detalladamente la transformación de un MER según las reglas para las entidades y se expresa en esquema relacional y modelo relacional.

NombreTabla(Identificador, Atributo1, Atributo2, Candidata)

#### 3.1 Transformación para las entidades y atributos

- Entidades. Las entidades pasan a ser tablas
- Atributos. Los atributos pasan a ser columnas o atributos de la tabla.
- Identificadores principales. Pasan a ser claves primarias
- Identificadores candidatos. Pasan a ser claves candidatas.

La siguiente imagen ilustra los anteriores elementos:

EMP_ID	EMP_MOMBRE	EMP_CARGO	EMP_JEFE	DEP_ID	
1000	LOPEZ	PRESIDENTE		10	Campo Nulo
1100	BENAVIDEZ	GERENTE	1000	30	
1200	CORDOBA	GERENTE	1000	10	
1300	JIMENEZ	GERENTE	1000	20	
2100	MARTINEZ	VENDEDOR	1100	30	
2200	ARTEAGA	VENDEDOR	1100	30	Filo / Tupla
2300	TOLEDO	VENDEDOR	1100	30	
2400	JARAMILLO	OFICINISTA	1100	30	Celda / Campo
2500	ZAMBRANO	VENDEDOR	1100	30	

Llave Primaria PK      Columna Atributo      Llave Foranea FK

Por ejemplo:

Si se tiene la siguiente entidad Aprendiz se convierte entonces:

<b>MER</b>	
<b>Esquema Relacional</b>	<p>TblAprendiz(AprCedula, AprNombre, AprApellido, AprTelefono, AprEmail)</p> <p>En este esquema la llave primaria se identifica porque se encuentra subrayado el atributo que será llave primaria</p>
<b>Modelo Relacional</b>	<div> <div>TblAprendiz</div> <div> <div>  AprCedula: INTEGER           </div> <div>  AprNombre: VARCHAR           </div> <div>  AprApellido: VARCHAR           </div> <div>  AprTelefono: INTERGER           </div> <div>  Apremail: VARCHAR           </div> </div> </div> <p>La tabla se llama TblAprendiz y la llave primaria se identifica porque el nombre puede estar: subrayado, tener al frente el estereotipo &lt;&lt;pk&gt;&gt; o una llave en color amarillo.</p>

**Nota:** para garantizar unicidad de estilos recuerde que debe tener una estandarización para nombrar las tablas y los atributos. Por ejemplo para este caso la entidad aprendiz se convierte en la tabla aprendiz y es llamada como TblAprendiz, el estilo de nombrado es iniciar con las tres primeras letras Tbl para indicar que es tabla, seguido del nombre de la entidad e iniciando con la primera letra en mayúscula.

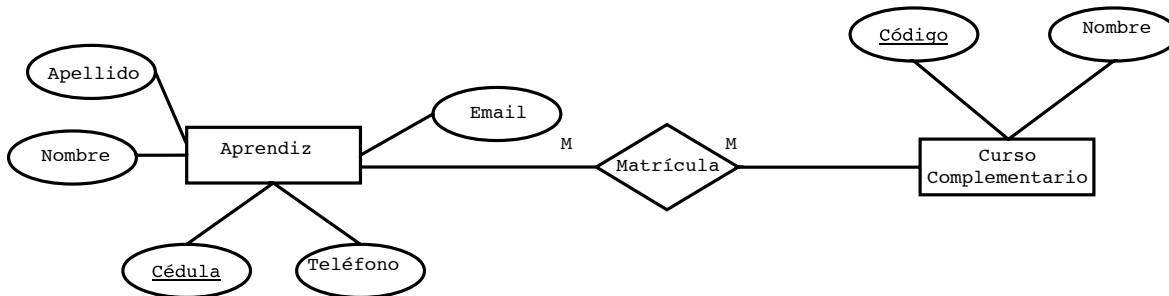
Para el atributo Apellido, en el modelo relacional se convierte en AprApellido indicando Apr que viene de la tabla Aprendiz y seguido del nombre del atributo cada uno iniciando con Mayuscula la primera letra.

### 3.2 Transformación para las Relaciones

#### 3.2.1 Relación Muchos a Muchos (M..M):

Esta relación se distingue en un MER porque su cardinalidad entre las dos entidades esta designada por (M..M) o (N..M).

Ejemplo:

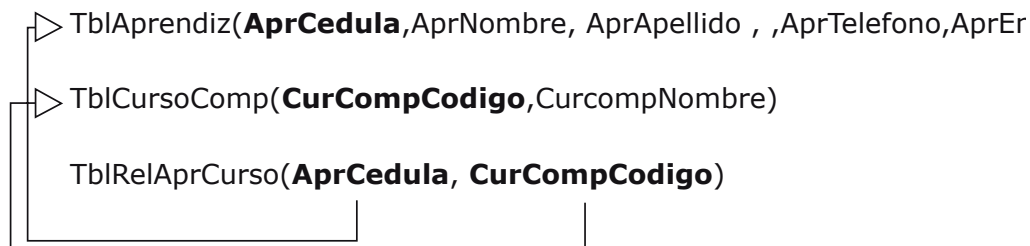


#### Transformación Relación Muchos a Muchos (M..M):

Cuando se tiene una relación de este tipo se debe crear una nueva relación (tabla) la relación se transforma en una tabla cuyos atributos son: los atributos de la relación y las claves de las entidades relacionadas (que pasarán a ser claves externas). La clave de la tabla la forman todas las claves externas.

## Transformación a Esquema Relacional

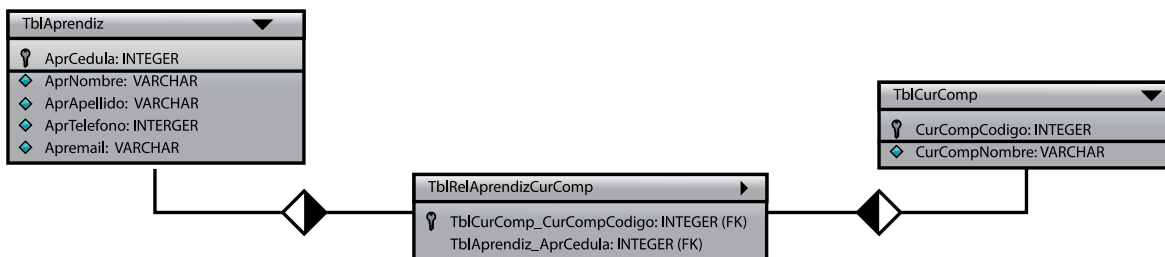
Cuando se hace la transformación a esquema relacional se debe tener un esquema de nombrado para las tablas que se generaran, por ejemplo para este caso la entidad aprendiz la tabla generada se denomina TblAprendiz, anteponiéndole tres letras Tbl que indican que es una tabla; de la misma manera para los atributos de esta entidad se le antepone las tres letras Apr para indicar que son atributos de la tabla Aprendiz. Así se continuara el estilo de nombrado para todos los casos de estudio y ejemplos de este material.



## Transformación a Modelo Relacional

Para realizar las transformaciones a modelo relacional puede utilizar herramientas case, para estos ejemplos se ha utilizado DbDesigner, el uso de ésta última se detallará en los videotutoriales que acompañan el laboratorio correspondiente.

Se visualiza la nueva tabla donde las llaves primarias son a la vez llaves foráneas, nótese que la cardinalidad de la relación está representada por los rombos en blanco (1) o sombreados (M)



La nueva tabla creada es llamada TblRelAprendizCurComp, que tiene incluida las dos llaves primarias de la tabla TblAprendiz y TblCurComp. Estas llaves en la nueva tabla son llamadas llaves foráneas y son representadas por un <fk> en el modelo.

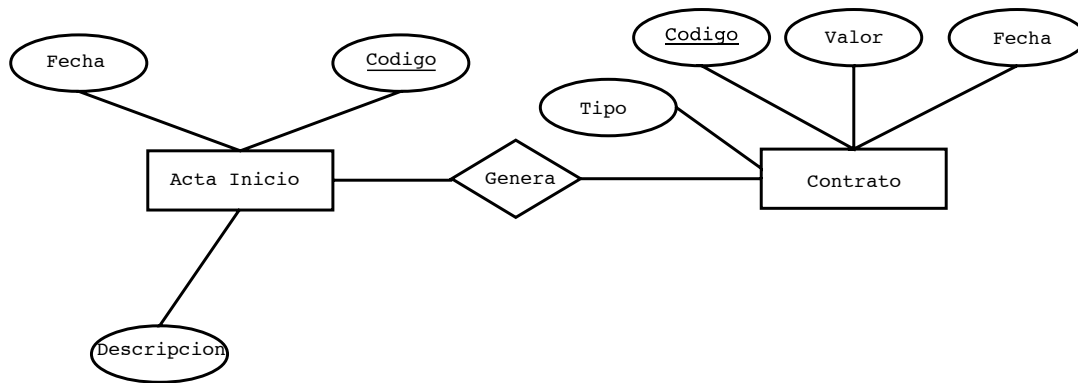


**Nota:** las llaves primarias pasadas a la nueva tabla se llaman llaves foráneas.

### 3.2.2 Relación 1 a 1 (1..1):

Esta relación se distingue en un MER porque su cardinalidad entre las dos entidades está designada por (1..1).

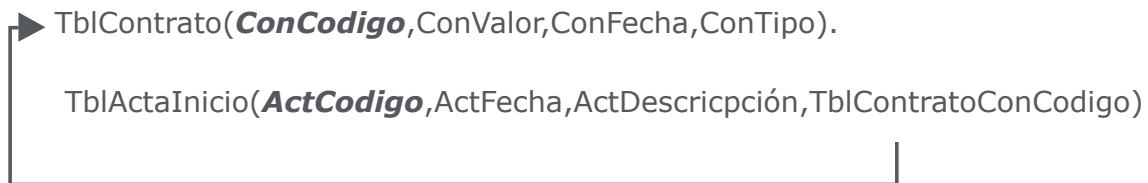
Ejemplo:



### Transformación Relación 1 a 1 (1..1):

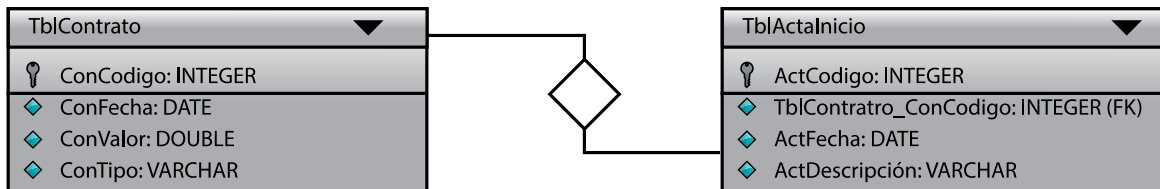
Cuando se tiene una relación de este tipo se envía la llave primaria de una tabla a la otra tabla que participa de la relación, se envía la llave primaria que se considere sea más relevante para enviar a la otra tabla.

### Esquema Relacional



Para este ejemplo específico se ha pasado la llave primaria del contrato (ConCodigo) a la tabla ActaInicio.

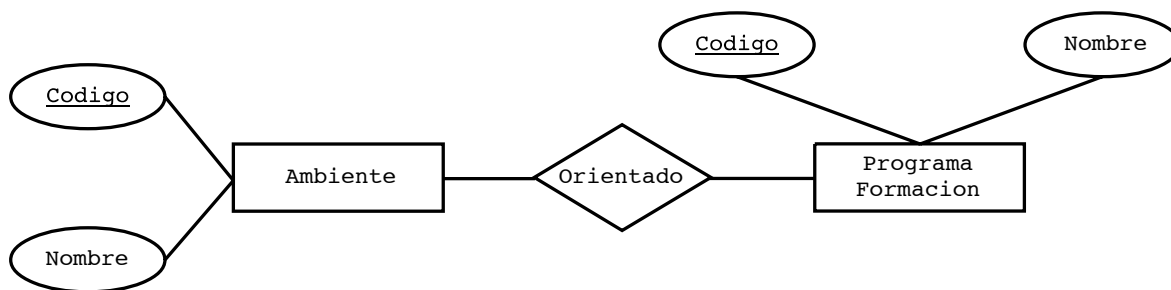
## Transformación a Modelo Relacional



En la tabla TblActaInicio se incluye la clave primaria de la tabla TblContrato (ConCodigo) quedando como llave foránea.

### 3.2.3 Relación 1 a Muchos (1..M):

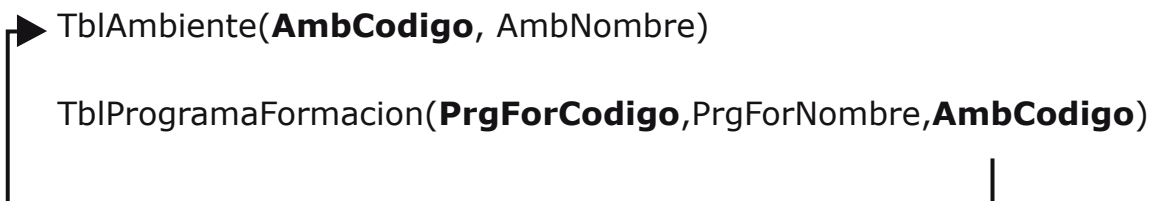
Esta relación se distingue en un MER porque su cardinalidad entre las dos entidades esta designada por (1..M).



### Transformación Relación 1 a Muchos (1..M):

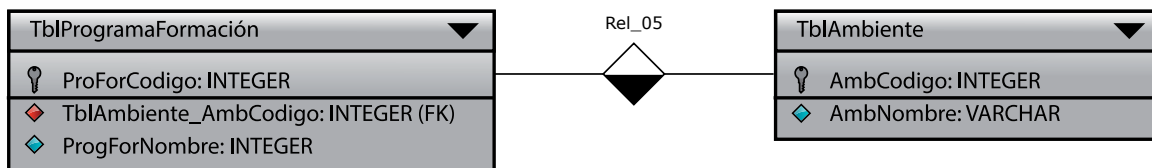
Cuando se tiene una relación de este tipo se debe colocar o enviar la llave primaria de la tabla que tiene en cardinalidad 1 a la tabla que tiene cardinalidad M

### Esquema relacional



Visualice que La tabla TblAmbiente tiene como cardinalidad 1, por la tanto la clave primaria de esta tabla es pasada a la tabla TblProgramaFormacion.

### Transformación a Modelo Relacional



Note que la clave de la tabla TblAmbiente es pasada a la Tabla TblProgramaFormacion la cual tiene su estereotipo <fk>.

## 4. TRANSFORMACIÓN MER- EXTENDIDO A RELACIONAL

A continuación se presenta un ejemplo de transformación de modelo entidad relación a Relacional, que aplica las reglas descritas. Para este ejemplo se toma una situación problemática y su modelo conceptual, para lo cual se presenta la descripción del problema, el MER y el modelo relacional resultante.

### Descripción del Problema

En el SENA Regional Cauca del Centro de Teleinformática y Producción industrial el coordinador requiere un sistema de información que permita gestionar los horarios de los instructores pertenecientes a un centro de formación que orientan formación profesional integral, tanto de planta como de contrato, tenga en cuenta que un instructor de contrato puede tener su relación de contrato por horas o asignación mensual.

Para lo cual es necesario conocer: Un instructor orienta formación profesional (clase) en un programa de formación (el programa de formación se refiere a las tecnologías o tecnólogos que ofrece el SENA), o en un curso complementario (los cuales son cursos de duración de 2 a 3 meses y solicitados por empresas del sector para capacitar a sus empleados).

Ambos: Programa de formación y complementarios son identificados por un código.

Cuando un aprendiz se matricula a un programa de formación o curso complementario tiene que durante toda su etapa lectiva (clase), cursar competencias laborales.

Una competencia podría asemejarse como una asignatura que un aprendiz debe cursar, pero para este caso siempre manejaremos el término competencia.

**Asignación de Competencias:** Para asignar una competencia a un instructor es necesario que esté legalmente contratado por el centro o que tenga resolución de nombramiento y según su especialidad se le asignan las competencias que va a orientar.

**Ambientes de formación:** Es el sitio específico (salón) donde se orientaran las competencias asignadas. Cada ambiente tiene una codificación asignada.

**Reglas del negocio:** Un instructor de contrato puede iniciar labores una vez tenga el acta de inicio.

Tenga en cuenta que un instructor puede laborar de lunes a viernes de 7:00 a 10:00 p.m. de la noche, sábados y domingos de 8:00 a 6:00 p.m.

Un instructor puede orientar diferentes competencias en diferentes programas de formación y cursos complementarios.

Un instructor no puede orientar una competencia en diferentes ambientes.

Dos o más instructores pueden orientar una o más competencias en un mismo ambiente.

**Usuarios:** Los usuarios que interactúan con la base de datos

Coordinador académico

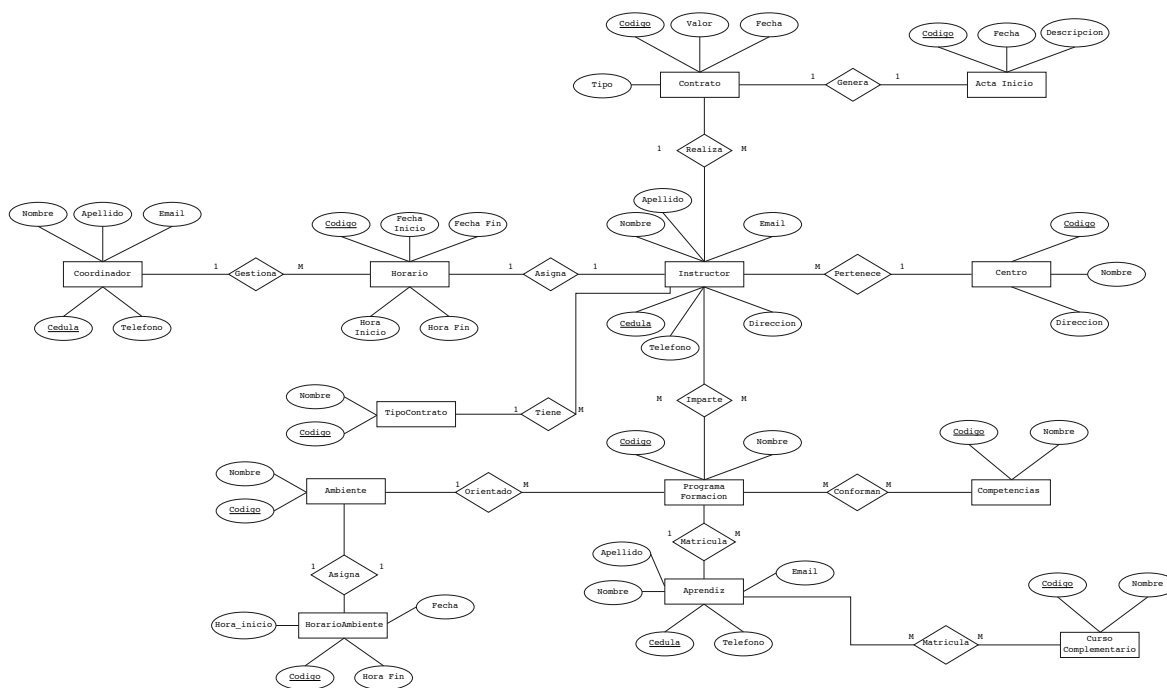
Instructores del SENA centro de Teleinformática y PI Regional Cauca

Informes por Usuario: Los informes necesarios para mostrar a los usuarios identificados son:

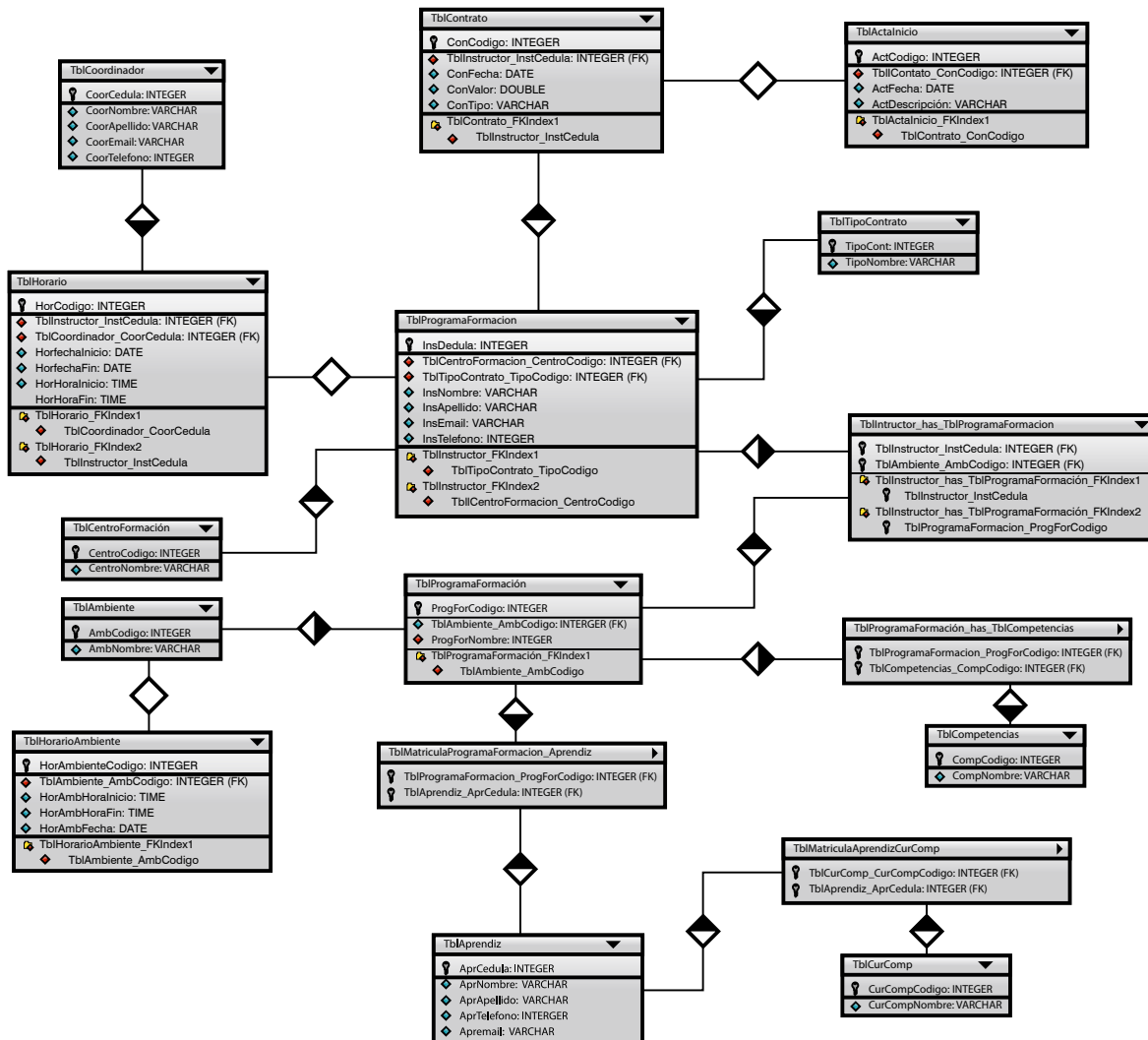


- Visualizar el horario de todos los instructores de planta.
- Visualizar el horario de todos los instructores de contrato
- Visualizar el horario de un solo instructor.
- Visualizar el horario de todos los ambientes
- Visualizar el horario de un solo ambiente.
- Visualizar el horario de todos los programas de formación.

### Modelo Entidad Relación



## Modelo Relacional



## 5. CONSTRUCCIÓN TRANSFORMACIÓN MER- EXTENDIDO A RELACIONAL

El MER-Extendido ha sido creado para modelar otros tipos de relaciones que no contempla el MER tradicional, los cuales son la generalización y especialización, recursivas, inclusividad y excursividad. A continuación se muestra como se transforma cada una de estas relaciones a un modelo relacional.

## 5.1 Relación especialización y generalización

Recordemos que en un MER-Extendido la generalización puede identificarse como solapamiento, exclusividad, parcialidad o totalidad Figura 2, en estas relaciones se encuentran entidad superclase o supertipos y entidades subclase o subtipos,

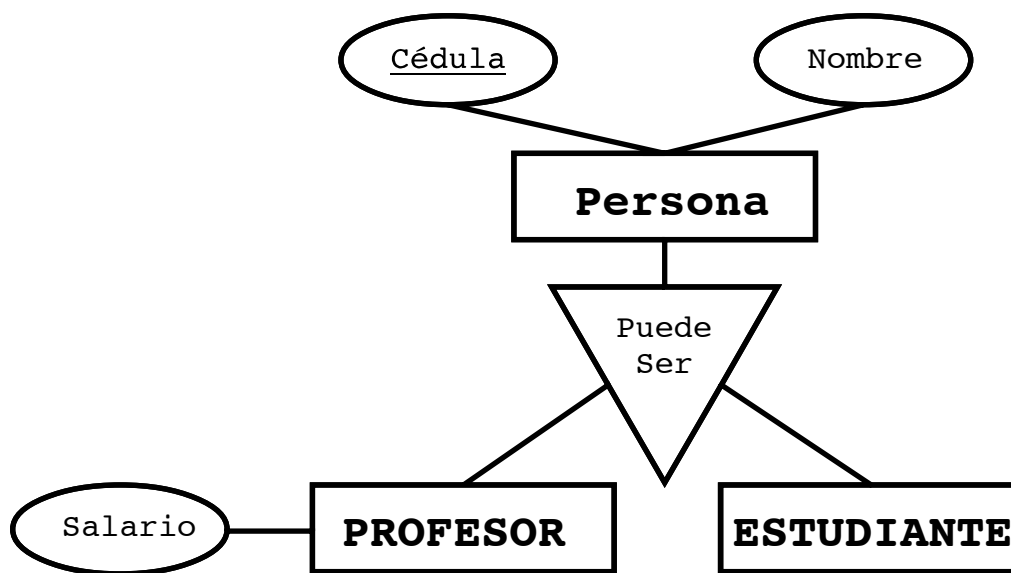


*Tipos de relación Generalización en un Mer Extendido*

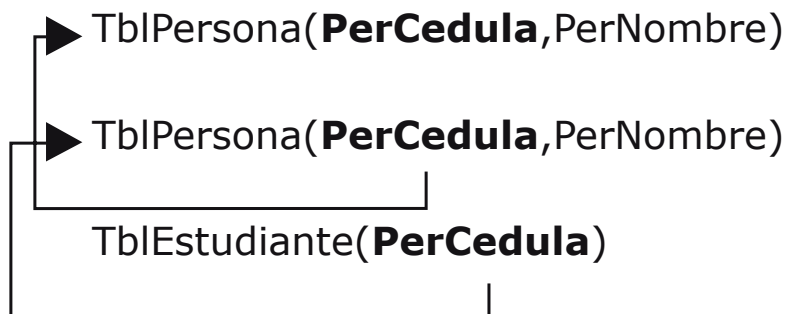
Existen cuatro opciones para la transformación de estos tipos de relaciones

**1. Si es solapada y parcial:** Crear una relación (**tabla**) para la superclase, con sus atributos correspondientes y una relación (**tabla**) para cada subclase con sus atributos incluyendo la clave primaria de la superclase.

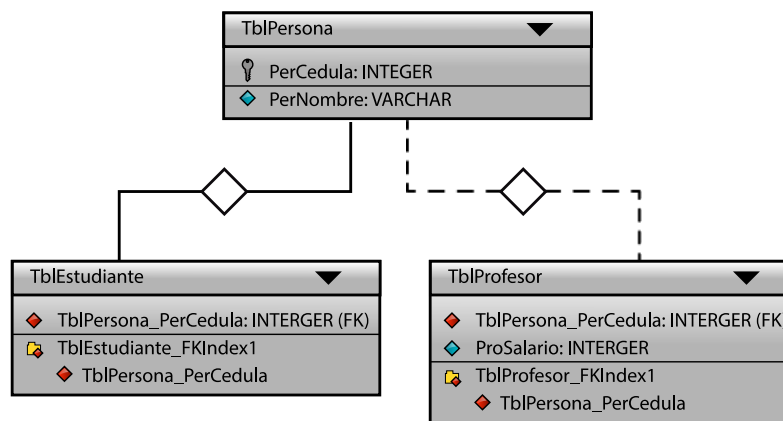
**Generalización:** esta representación es solapada y parcial, es decir una persona puede ser profesor y estudiante a la vez y no es obligatorio que asuma un subtipo.



### Transformación Esquema Relacional



### Transformación Modelo Relacional

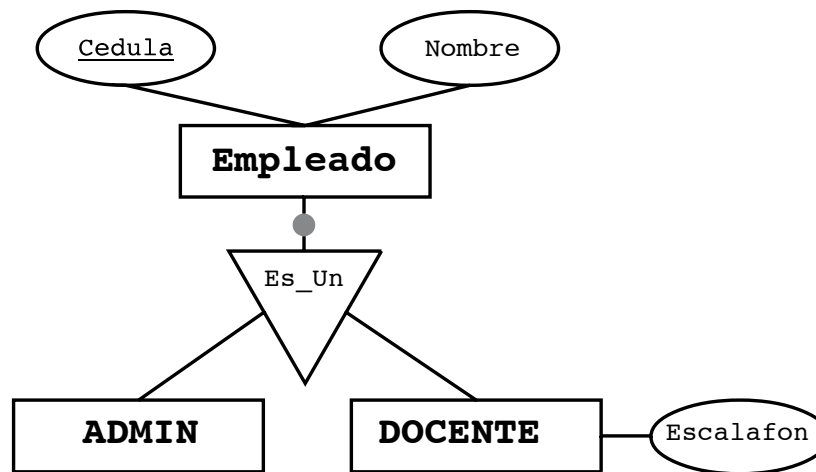




Observe que se creó la tabla persona con sus campos y se crearon las tablas profesor y estudiante que heredan la llave primaria de la tabla persona (PerCedula) y se convierte en una llave foránea.

**2. Si es total:** Se crea una relación (tabla) para las entidades subtipo, donde cada una tiene los atributos del supertipo. No se crea una relación para el supertipo.

**Relación de totalidad:** Obligatoriamente el empleado tiene que tomar algún subtipo o administrador y/o docente.



### Transformación Esquema Relacional

TblAdmin(AdminCedula,AdminNombre)

TblDocente(DocCedula,DocNombre)

### Transformación Modelo Relacional

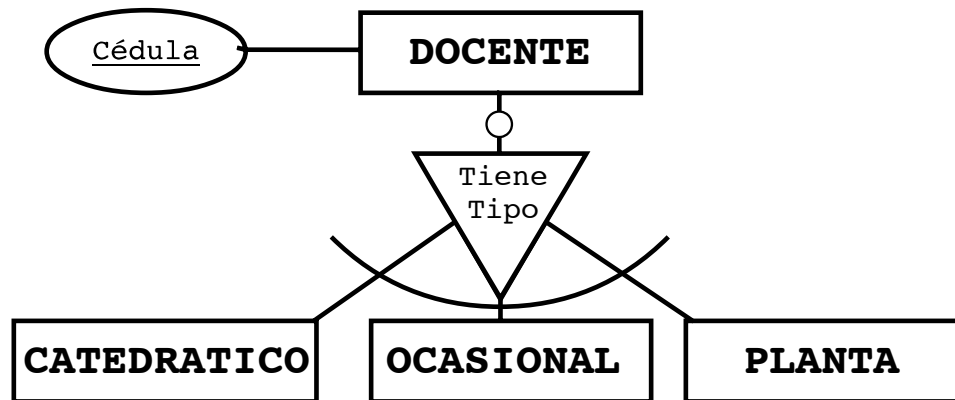
TblAdmin
AdminCedula: INTEGER
AdminNombre: VARCHAR

TblDocente
AdminCedula: INTEGER
AdminNombre: VARCHAR
DocEscalafo: INTEGER

Observe que quedan dos tablas y se suprime la superclase (empleado), cada una de las tablas hereda los atributos de la superclase y continúan con los atributos propios.

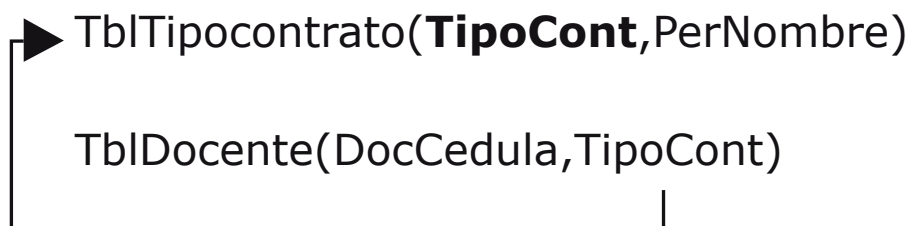
3. Crear una relación (tabla) para almacenar los subtipos y la relación (tabla) superclase. Teniendo en cuenta si es parcial, total, solapado o de exclusividad; Miremos los dos casos que pueden darse en este sentido:

**Caso 1:** Si la descripción del problema indica que obligatoriamente **pertenece a un solo tipo** es decir la relación es (total y exclusiva), entonces se crea una relación para la SuperClase y otra para el tipo y se le pasa la llave primaria de la tabla tipo a la tabla superclase.



En esta representación el docente puede tener un solo contrato (ocacional, catedrático y planta) y obligatoriamente tiene que tener asociado un contrato.

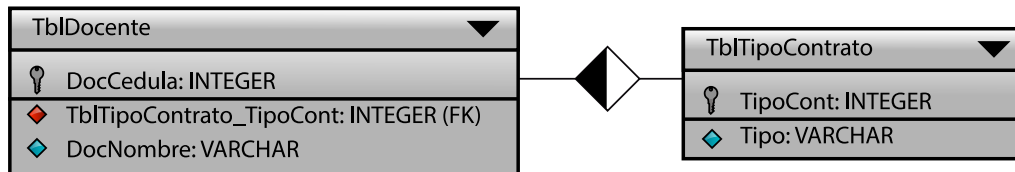
### Transformación esquema relacional



Observe que se crea una relación (tabla) que almacena los diferentes tipos de contrato que un docente puede tener y adicionalmente a la tabla TblDocente se le pasa la llave primaria de la tabla TipoContrato.

De esta manera se garantiza que cada docente obligatoriamente tenga un solo tipo de contrato.

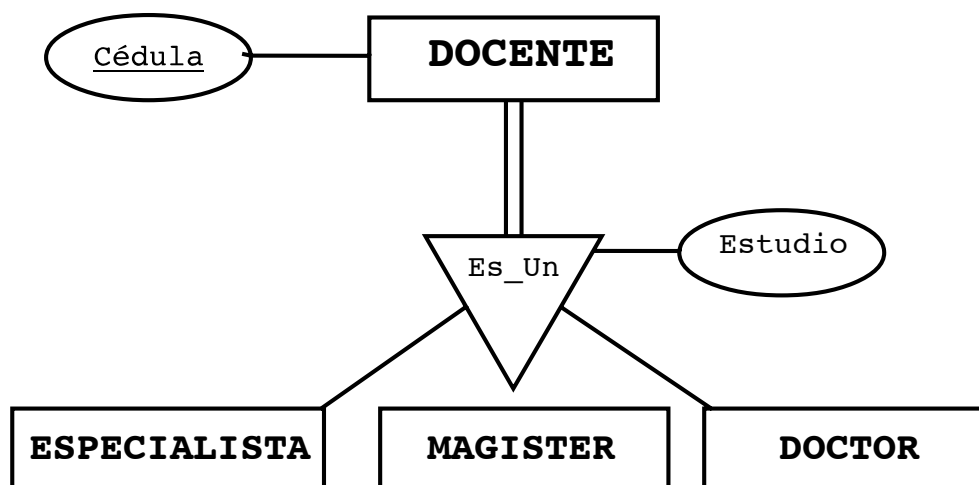
### Transformación modelo relacional



Observe que se genera una nueva tabla llamada TblTipCont que almacenara los diferentes tipos de contrato que puede tener un docente, para mayor explicación esta tabla se almacenara con los siguientes datos.

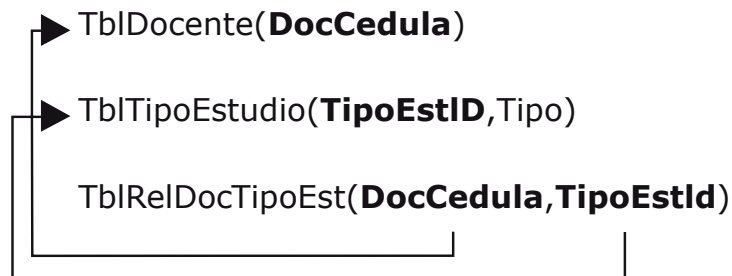
TblTipCont	
TipoContID	Tipo
1	Catedratico
2	Ocasional
3	Planta

**Caso 2:** Si la descripción del problema indica que puede **pertenecer a varios tipos** es decir la relación es (parcial y solapada), entonces se crea una relación para la SuperClase, otra para el tipo y se trata la relación superclase y Tipo como una relación de muchos a muchos.

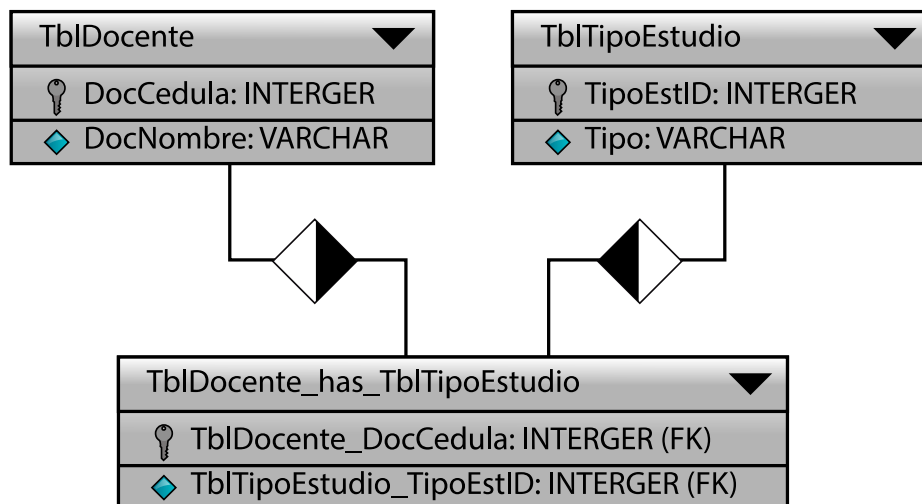


En esta representación un docente puede tener o no un tipo de estudio y puede tenerlos todos a la vez.

## Transformación Esquema Relacional

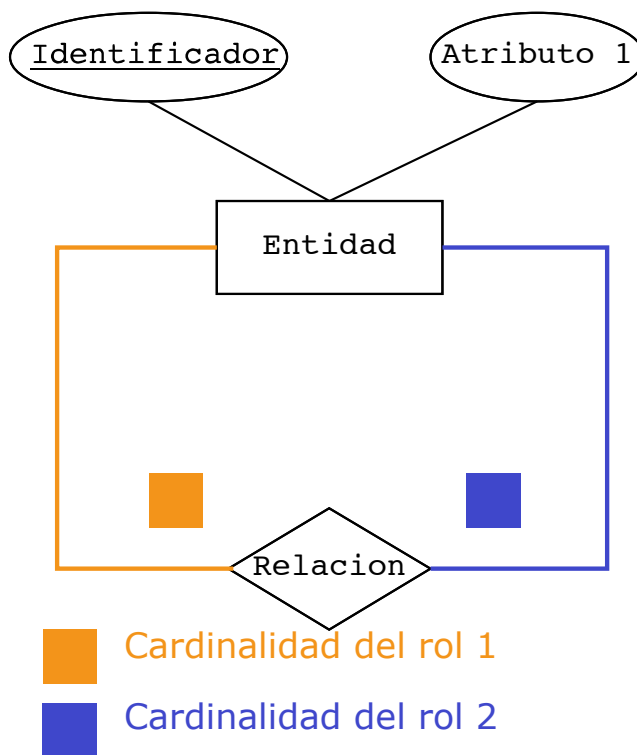


Observe que se crea una relacion (tabla) para la superclase y otra para el tipo y se tratan estas dos relaciones como muchos a muchos, por lo tanto se crea una tabla intermedia que contenga las llaves primarias de las 2 tablas creadas.

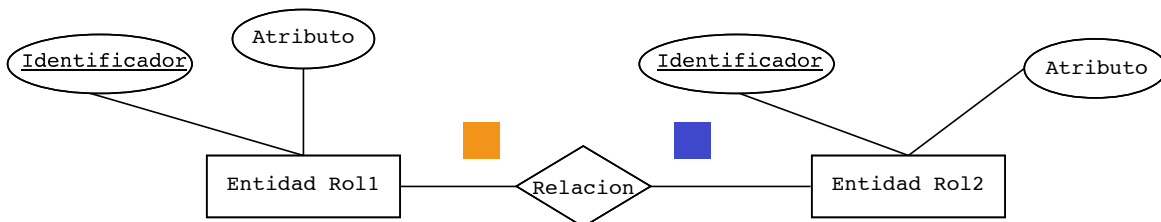


## 5.2 Relación Recursiva

Las relaciones recursivas se tratan de la misma forma que las otras, sólo que hay que imaginar que la tabla se divide en dos, una por cada rol. Teniendo en cuenta eso, la solución es idéntica a lo ya resuelto en los casos anteriores.

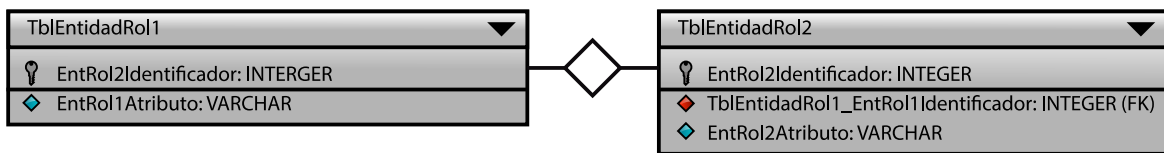


La forma de imaginarse las relaciones recursivas para resolverlas es:



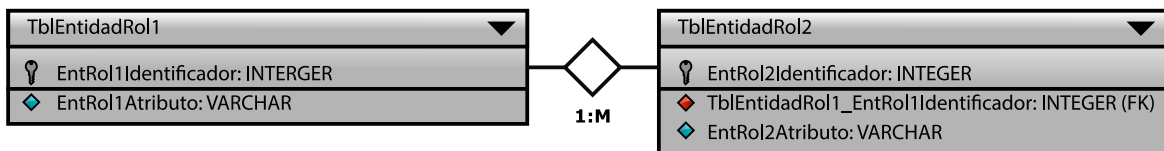
Donde se "duplica la entidad" y se conservan las cardinalidades, luego dependiendo de ésta, se aplica la regla de transformación correspondiente teniendo las siguientes posibilidades:

1 caso: si la cardinalidad de la relación es de 1 a 1:

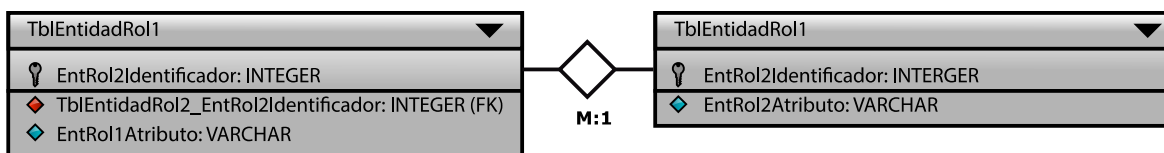


La entidad con Rol débil hereda el atributo de identificación de la otra entidad y pasa a ser su llave foránea

2 caso: si la cardinalidad de la relación es de 1 a M :

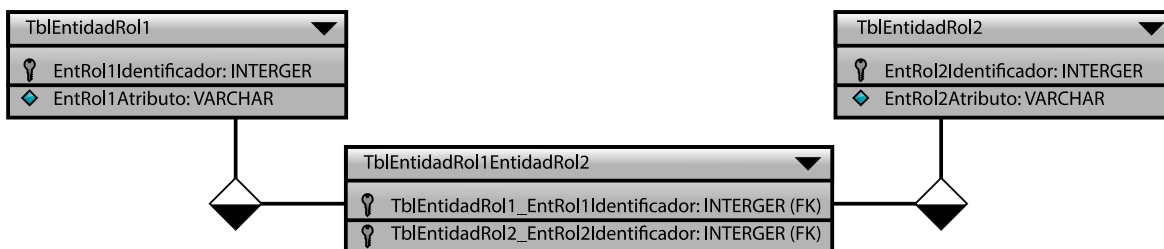


O de M a 1:



En cualquiera de los casos, la entidad con relación muchos hereda el atributo de identificación de la otra y esta se convierte en la llave foránea

3 caso: si la cardinalidad de la relación es de M a M:



Se genera en este caso una tercera tabla que relaciona las dos entidades, está compuesta por los atributos de identificación que hereda de cada entidad, más los atributos propios de la relación (si existieran).

## 6. Normalización

Una vez finalizado el proceso de conversión, se pueden presentar ciertos inconvenientes tales como:

- Incapacidad de almacenar ciertos hechos.
- Redundancias y por tanto posibilidad de inconsistencias.
- Ambigüedades
- Pérdida de información
- Pérdida de ciertas restricciones de integridad que dan lugar a interdependencia de los datos.
- Existencia de valores nulos.
- Aparición de estados inválidos en los proceso de manipulación.

La normalización es el proceso mediante el cual se transforman datos complejos a un conjunto de estructuras de datos más pequeñas, que además de ser más simples y más estables, son más fáciles de mantener. También se puede entender la normalización como una serie de reglas que sirven para ayudar a los diseñadores de bases de datos a desarrollar un esquema que minimice los problemas de lógica. Cada regla está basada en la que le antecede.

La normalización también hace las cosas fáciles de entender. Los seres humanos tenemos la tendencia de simplificar las cosas al máximo. Lo hacemos con casi todo, desde los animales hasta con los automóviles. Vemos una imagen de gran tamaño y la hacemos más simple agrupando cosas similares juntas. Las guías que la normalización provee crean el marco de referencia para simplificar una estructura de datos compleja.

Otra ventaja de la normalización de base de datos es el consumo de espacio. Una base de datos normalizada ocupa menos espacio en disco que una no normalizada. Hay menos repetición de datos, lo que tiene como consecuencia un mucho menor uso de espacio en disco.

Una vez obtenido el esquema relacional resultante del esquema entidad/relación que representa la base de datos, normalmente tendremos una buena base de datos. Pero otras veces, debido a fallos en el diseño o a problemas indetectables, tendremos un esquema que puede producir una base de datos que incorpore estos problemas:



**Redundancia:** Se llama así a los datos que se repiten continua e innecesariamente por las tablas de las bases de datos. Cuando es excesiva es evidente que el diseño hay que revisarlo, es el primer síntoma de problemas y se detecta fácilmente.

**Ambigüedades:** Datos que no clarifican suficientemente el registro al que representan. Los datos de cada registro podrían referirse a más de un registro o incluso puede ser imposible saber a qué ejemplar exactamente se están refiriendo. Es un problema muy grave y difícil de detectar.

**Perdida Restricciones de Integridad:** Normalmente debido a dependencias funcionales. Más adelante se explica este problema. Se arreglan fácilmente siguiendo una serie de pasos concretos.

**Anomalías en operaciones de modificación de datos:** El hecho de que al insertar un solo elemento haya que repetir tuplas en una tabla para variar unos pocos datos. O que eliminar un elemento suponga eliminar varias tuplas necesariamente (por ejemplo que eliminar un cliente suponga borrar seis o siete filas de la tabla de clientes, sería un error muy grave y por lo tanto un diseño terrible).

El principio fundamental reside en que las tablas deben referirse a objetos o situaciones muy concretas, relacionados exactamente con elementos reconocibles por el sistema de información de forma inequívoca. Cada fila de una tabla representa inequívocamente un elemento reconocible en el sistema. Lo que ocurre es que conceptualmente es difícil agrupar esos elementos correctamente.

En cualquier caso la mayor parte de problemas se agravan si no se sigue un modelo conceptual y se decide crear directamente el esquema relacional. En ese caso, el diseño tiene una garantía casi asegurada de funcionar mal.

Cuando aparecen los problemas enumerados, entonces se les puede resolver usando reglas de normalización. Estas reglas suelen forzar la división de una tabla en dos o más tablas para arreglar ese problema.

### 6.1 Grados de Normalización

Las formas normales corresponden a una teoría de normalización iniciada por el propio Codd continuada por otros autores (entre los que destacan



Boycey Fagin). Codd definió en 1970 la primera forma normal, desde ese momento aparecieron la segunda, tercera, la Boyce-Codd, la cuarta y la quinta forma normal.

Una tabla puede encontrarse en primera forma normal y no en segunda forma normal, pero no al contrario. Es decir los números altos de formas normales son más restrictivos (la quinta forma normal cumple todas las anteriores).

### 6.1.1 Primera forma Normal (1FN)

Es una forma normal inherente al esquema relacional. Es decir toda tabla realmente relacional la cumple.

Se dice que una tabla se encuentra en primera forma normal si impide que un atributo de una tupla pueda tomar más de un valor. La tabla:

EMPLEADO		
Identificación	nombre	departamento
12345	Pedro Murcia	Sistemas
24680	Maria Dolores Rosas	Contabilidad
12232	Juan José Muñoz	Gerencia Recursos Humanos

Lo que vemos es una tabla, pero no una tabla relacional (lo que en terminología de bases de datos relacionales se llama relación). No cumple la primera forma normal.

Podríamos mejorar para que cumpla la primera forma normal así:

Asignación de Competencias:

EMPLEADO		
Identificación	nombre	departamento
12345	Pedro Murcia	Sistemas
24680	María Dolores Rosas	Contabilidad
12232	Juan José Muñoz	Gerencia
12232	Juan José Muñoz	Recursos Humanos

### 6.1.2 Segunda forma Normal (2FN)

Si una tabla está en primera forma normal y además cada atributo que no sea clave, depende de forma funcional completa respecto de cualquiera de las claves. Toda la clave principal debe hacer dependientes al resto de atributos, si hay atributos que depende sólo de parte de la clave, entonces esa parte de la clave y esos atributos formarán otra tabla.

Para asignar una competencia a un instructor es necesario que esté legalmente contratado por el centro o que tenga resolución de nombramiento y según su especialidad se le asignan las competencias que va a orientar.

Alumno				
identificacion	codCurso	nombre	apellido	nota
11	34	Sofia	Loren	5
11	25	Sofía	Loren	4
12	34	Marcos	Pinto	3.5
13	25	Mireya	Ruano	4.5
13	34	Mireya	Ruano	5

Suponiendo que la identificación y el código del curso formen una clave principal para esta tabla, sólo la nota tiene dependencia funcional completa. El nombre y el apellido dependen de forma completa de la identificación. La tabla no es 2FN, para convertirla debemos convertir esa tabla en dos tablas así:

Alumno		
identificacion	nombre	apellido
11	Sofia	Loren
11	Sofía	Loren
12	Marcos	Pinto
13	Mireya	Ruano
13	Mireya	Ruano
NotasCurso		
identificacion	codCurso	nota

11	34	5
11	25	4
12	34	3.5
13	25	4.5
13	34	5

### 6.1.3 Tercera forma Normal (3FN)

Ocorre cuando una tabla está en 2FN y además ningún atributo que no sea clave depende transitivamente de las claves de la tabla. Es decir no ocurre cuando algún atributo depende funcionalmente de atributos que no son clave.

Ejemplo:

ALUMNOS				
Identificación	nombre	apellido	codCiudad	ciudad
31	Pedro	Picapiedra	100	Cali
32	Mónica	Galindo	100	Cali
33	Amparo	Grisales	101	Popayán
34	Roberto	Gómez	102	Neiva
35	Blanca	Moreno	102	Neiva

En la tabla anterior la ciudad depende funcionalmente del código de la ciudad, lo que hace que no esté en 3FN. La manera de solucionar el problema sería dividiendo la tabla en dos así:

En la primera tabla creamos la tabla Alumnos con los atributos propios de él y solo le agregamos el atributo de codCiudad para saber a que ciudad pertenece.

ALUMNOS

Identificación	Nombre	Apellido	codCiudad
31	Pedro	Picapiedra	100
32	Mónica	Galindo	100
33	Amparo	Grisales	101

34	Roberto	Gómez	102
35	Blanca	Moreno	102

La segunda tabla corresponde a la ciudad con sus atributos.

CIUDAD	
codCiudad	Ciudad
100	Cali
100	Cali
101	Popayán
102	Neiva
102	Neiva

#### 6.1.4 Forma normal de Boyce-Cood(FNBC o BCFN)

Ocurre si una tabla está en tercera forma normal y además todo determinante es una clave candidata. Ejemplo:

EMPRESA		
Empleado	Departamento	Responsable
Pedro	Dirección	Juan
Mari	Ventas	Luis
Jose	Mercadeo	Erika
Rosa	Ventas	Roció
Blanca	Gerencia	Andrés
Julian	Ventas	Mireya

La cuestión es que un empleado puede trabajar en varios departamentos. En cada departamento hay varios responsables, pero cada empleado sólo tiene asignado uno. El detalle importante que no se ha tenido en cuenta, es que el o la responsable sólo puede ser responsable en un departamento. Este detalle último produce una dependencia funcional ya que:

Responsable    Departamento

Por lo tanto hemos encontrado un determinante que no es clave candidata. No está por tanto en FNBC. En este caso la redundancia ocurre por mala

selección de clave. La redundancia del departamento es completamente evitable. La solución sería:

PERSONAL	
Empleado	Responsable
Pedro	Juan
Mari	Luis
Jose	Erika
Rosa	Roció
Blanca	Andrés
Julian	Mireya

RESPONSABLES	
Responsable	Departamento
Juan	Dirección
Luis	Ventas
Erika	Mercadeo
Roció	Ventas
Andrés	Gerencia
Mireya	Ventas

En las formas de Boyce-Codd hay que tener cuidado al descomponer ya que se podría perder información por una mala descomposición.

### 6.1.5 Cuarta Forma Normal (4FN) Dependencias Multievaluadas.

#### Dependencia Multivaluada

Para el resto de formas normales (las diseñadas por **Fagin**, mucho más complejas), es importante definir este tipo de dependencia, que es distinta de las funcionales. Si las funcionales eran la base de la segunda y tercera forma normal (y de la de Boyce-Codd), éstas son la base de la cuarta forma normal.

Una dependencia multivaluada de X sobre Y (es decir  $X \twoheadrightarrow Y$ ), siendo X e Y atributos de la misma tabla, ocurre cuando Y tiene un conjunto de valores

bien definidos sobre cualquier valor de X. Es decir, dado X sabremos los posibles valores que puede tomar Y.

Se refiere a posibles valores (en plural) y se trata de que los valores de ese atributo siempre son los mismos según el valor de un atributo y no del otro.

### Ejemplo:

Código Curso	Instructor	Material
11	Marta	1
11	Marta	2
11	Jaime	1
11	Jaime	2
12	Marta	1
12	Marta	2
12	Marta	3

La tabla cursos, instructores y materiales del curso. La tabla está en FNBC ya que no hay dependencias transitivas y todos los atributos son clave sin dependencia funcional hacia ellos. Sin embargo hay redundancia. Los materiales se van a repetir para cualquier instructor dando cualquier curso, ya que los instructores van a utilizar todos los materiales del curso (de no ser así no habría ninguna redundancia).

Los materiales del curso dependen de forma multivaluada del curso y no del instructor en una dependencia multivaluada (no hay dependencia funcional ya que los posibles valores son varios). Para el **Código curso** e **Instructor** podemos saber los materiales; pero lo sabemos por el curso y no por el instructor.

### Cuarta Forma Normal

Ocurre esta forma normal cuando una tabla está en forma normal de BoyceCodd y toda dependencia multivaluada no trivial es una dependencia funcional. Son triviales aquellas dependencias multivaluadas en las que el conjunto formado por el determinante y el implicado no forman la clave primaria de la tabla y además el implicado no forma parte del determinante: es decir si  $X \twoheadrightarrow Y$  y además  $Y \not\subset X$  y  $X, Y$  no es la clave de

la tabla, tenemos una dependencia multivaluada no trivial (como ocurre en el ejemplo anterior).

Codigo Curso	Material
11	1
11	2
12	1
12	2
12	3

Codigo Curso	Instructor
11	Marta
11	Jaime
12	Marta

### 6.1.6 Quinta Forma Normal (5FN)

#### Dependencias de Join o de reunión

Una **proyección** de una tabla es la tabla resultante de tomar un subconjunto de los atributos de una. Es decir una tabla formada por unas cuantas columnas de la tabla original.

La operación **JOIN** procedente también del álgebra relacional, consiste en formar una tabla con la unión de dos tablas. La tabla resultante estará formada por la combinación de todas las columnas y filas de ambas, excepto las columnas y filas repetidas.

Se dice que se tiene una tabla con **dependencia de unión (o de tipo JOIN)** si se puede obtener esa tabla como resultado de combinar mediante la operación JOIN varias proyecciones de la misma.

#### Quinta forma normal o forma normal de proyección-uni3n

Ocurre cuando una tabla est1 en 4FN y cada dependencia de uni3n (JOIN) en ella es implicada por las claves candidatas.

Es la más compleja y polémica de todas. Polémica pues no está claro en muchas ocasiones está muy claro que el paso a 5FN mejore la base de datos. Fue definida también por Fagin. Es raro encontrarse este tipo de problemas cuando la normalización llega a 4FN. Se deben a restricciones semánticas especiales aplicadas sobre la tabla.

Ejemplo:

Proveedor	Material	Proyecto
1	1	2
1	2	1
2	1	1
1	1	1

Indican códigos de material suministrado por un proveedor y utilizado en un determinado proyecto. Así vista la tabla, no permite ninguna proyección en la que no perdamos datos. Pero si ocurre una restricción especial como por ejemplo: **Cuando un proveedor nos ha suministrado alguna vez un determinado material, si ese material aparece en otro proyecto, haremos que el proveedor anterior nos suministre también ese material para el proyecto.**

Eso ocurre en los datos como el proveedor número 1 nos suministró el material número 1 para el proyecto 2 y en el proyecto 1 utilizamos el material 1, aparecerá la tupla proveedor 1, material 1 y proyecto 1. Si un nuevo proyecto necesitara el material 1, entonces habrá que pedirlo a los proveedores 1 y 2 (ya que en otros proyectos los hemos utilizado)

La dependencia de reunión que produce esta restricción es muy difícil de ver ya que es lejana. Para esa restricción esta proyección de tablas sería válida:

Proveedor	Material
1	1
1	2
2	1



Proveedor	Proyecto
1	2
1	1
2	1

Material	Proyecto
1	2
2	1
1	1

Esa descomposición no pierde valores en este caso, sabiendo que si el proveedor nos suministra un material podremos relacionarle con todos los proyectos que utilizan ese material. Una tabla no está en quinta forma normal si hay una descomposición de esa tabla que muestre la misma información que la original y esa descomposición no tenga como clave la clave original de la tabla.

### Forma Normal de Dominio Clave (FNDC)

Se la conoce más con sus siglas en inglés **DKNF**. Se trata de una forma normal enunciada también por Fagin en 1981 al darse cuenta de los problemas de redundancia que ocurrían con algunos dominios. En este caso no se basó en dependencias entre los datos, sino que se basó en **restricciones de dominio** y **restricciones de clave**.

**Restricciones de dominio:** Se trata de la restricción que hace que un determinado atributo obtenga sólo ciertos valores, los que estén de acuerdo a la definición de un determinado dominio.

**Restricción de clave:** Es la restricción que permite que un atributo o un conjunto de atributos forme una clave candidata.

Fagin dice que una tabla está en **FNDC** si toda restricción sobre la tabla es consecuencia lógica de aplicar las restricciones de dominio y clave sobre la misma. Fagin demostró que si esto ocurría la tabla incluso estaba en 5FN.

Ejemplo:

Alumno	Nivel Logrado	Nota
Rosa	Muy alto y eficiente	9
María	Muy alto aunque sin trabajo constante	9
Juan	Alto con trabajo habitual	7
José	Medio aunque con trabajo	5

Observando los datos de la tabla se observa que cuando la nota es igual o superior a 9, en el nivel aparece la palabra alto, cuando es un 7 o un 8 medio, y un 5 o 6 sería medio. Es decir tenemos restricciones que no son ni de dominio ni de clave en esa tabla. Lo lógico sería:

Alumno	Nivel Logrado	Nota
Rosa	Eficiente	9
María	Trabajo constante	9
Juan	Trabajo habitual	7
José	Trabajo Medio	5

Nivel Académico	Nota Mínima	Nota Máxima
Rosa	9	10
María	7	8,99
Juan	5	6,99
José	0	4,99

No se pierde información al diseñar las tablas de esta forma y de hecho es más eficiente para la base de datos.

## GLOSARIO

**Modelo Entidad Relación:** Modelo que consta de entidades, atributos y relaciones y permite modelar conceptualmente una base de datos relacional.

**Modelo Entidad Relación Extendido:** Modelo que permite modelar nuevos tipos de relaciones, que no incluye el MER normal, tales como agregación, generalización, exclusión e inclusión.

**Modelo Relacional:** Es un modelo que se obtiene de la transformación de un MER, el cual contiene tablas, atributos y relaciones entre las tablas.

**Tabla:** Es la representación de una entidad en un modelo relacional, la cual incluye atributos.

**Campo:** Es la representación de un atributo en un modelo relacional el cual contiene su nombre, tipo de dato y descripción de si es clave primaria o llave foránea.

**Clave Primaria:** Atributo único que no se repite, es identificado en un modelo relacional porque esta subrayado o tiene el estereotipo <pk> y en un esquema relacional se encuentra subrayado.

**Clave Foránea:** Atributo que es pasado a una tabla, según la regla de transformación, se distingue en un modelo relacional porque tiene el estereotipo <fk> y en un esquema relacional porque despliega una flecha hacia la tabla de la cual proviene.

**Relación:** Es una asociación que existe entre dos entidades, en un MER se distinguen tres tipos de relaciones: uno a uno (1..1), uno a muchos (1..M) y muchos a muchos (M..M).

**Esquema Relacional:** Expresar el modelo relacional por medio un esquema de palabras y asociaciones entre ellas (flechas) que representan las relaciones.

**1FN:** Abreviatura de Primera Forma Normal.

**2FN:** Abreviatura de Segunda Forma Normal.

**3FN:** Abreviatura de Tercera Forma Normal.

**4FN:** Abreviatura de Cuarta Forma Normal.

**5FN:** Abreviatura de Quinta Forma Normal.

**FNBC:** Abreviatura de Forma Normal de BoyceCodd.



## BIBLIOGRAFÍA

---

Universidad de los Andes. (5 de Diciembre de 2012). Nivelatorio de Bases de Datos. Recuperado el 4 de Julio de 2013, de Departamento de Ingeniería de Sistemas y Computación: [http://sistemas.uniandes.edu.co/~arti4002/dokuwiki/lib/exe/fetch.php?media\x3dprincipal:05\\_conversionmodeloconceptualarelacional.pptx](http://sistemas.uniandes.edu.co/~arti4002/dokuwiki/lib/exe/fetch.php?media\x3dprincipal:05_conversionmodeloconceptualarelacional.pptx)

Escuela Superior de Informática de Ciudad Real. (8 de Febrero de 2008). Itescam . Recuperado el 4 de Julio de 2013, de <http://www.itescam.edu.mx/principal/sylabus/fpdb/recursos/r88166.PDF>

Henry F. Korth , (2006). Fundamentos de Bases de Datos. McGraw-Hill.

Sánchez, J. (2004). Principios sobre bases de datos. Obtenido de <http://www.jorgesanchez.net/bd/bdrelacional.pdf>

Silberschatz, A. (2004). En e. Al, Fundamentos de Bases de Datos

Universidad de los Andes. (5 de Diciembre de 2012). Nivelatorio de Bases de Datos. Recuperado el 4 de Julio de 2013, de Departamento de Ingeniería de Sistemas y Computación: [http://sistemas.uniandes.edu.co/~arti4002/dokuwiki/lib/exe/fetch.php?media\x3dprincipal:05\\_conversionmodeloconceptualarelacional.pptx](http://sistemas.uniandes.edu.co/~arti4002/dokuwiki/lib/exe/fetch.php?media\x3dprincipal:05_conversionmodeloconceptualarelacional.pptx)



<b>Control de documento</b> <b>Construcción Objeto de Aprendizaje</b> <b>Conversión MER a Modelo Relacional</b>	
Desarrollador de contenido Experto temático	Leydy Carolina Muñoz Pachajoa Cesar Marino Cuellar
Asesor pedagógico	Rafael Neftalí Lizcano Reyes Claudia Milena Hernández Naranjo
Producción Multimedia	Luis Fernando Botero Mendoza Victor Hugo Tabares Carreño
Programadores	Daniel Eduardo Martínez
Líder expertos temáticos	Ana Yaqueline Chavarro Parra
Líder línea de producción	Santiago Lozada Garcés

